

Implementarea filtrelor digitale IIR în forma lattice

Laborator 6, PSS

Obiectiv

Familiarizarea studenților cu formele de implementare tip *lattice* pentru filtre IIR

Noțiuni teoretice

Exerciții

1. Fie sistemul IIR cauzal cu poli și zerouri, cu funcția de sistem:

$$H(z) = \frac{1 + 2z^{-1} + 3z^{-2} + 2z^{-3}}{1 + \frac{2}{5}z^{-1} + \frac{7}{20}z^{-2} + \frac{1}{2}z^{-3}}$$

Determinați și desenați structura echivalentă *lattice* cu poli și zerouri.

2. Se dă sistemul IIR cauzal numai cu poli, cu funcția de sistem:

$$H(z) = \frac{1}{1 + \frac{2}{5}z^{-1} + \frac{7}{20}z^{-2} + \frac{1}{2}z^{-3}}$$

Determinați coeficienții structurii *lattice* și desenați-o.

3. Utilizați utilitarul `fdatool` pentru a proiecta unul din filtrele următoare:
 - a. Un filtru trece-jos IIR de ordin 4, de tip eliptic, cu frecvența de tăiere de 5kHz la o frecvență de eșantionare de 44.1kHz;
 - b. Un filtru trece-sus IIR de ordin 4, de tip eliptic, cu frecvența de tăiere de 1kHz la o frecvență de eșantionare de 44.1kHz;

- c. Un filtru trece-bandă IIR de ordin 4, de tip eliptic, cu banda de trecere între 700Hz și 4kHz la o frecvență de eșantionare de 44.1kHz.
4. În mediul Simulink, realizați implementarea IIR filtrului de mai sus în forma *lattice*.
5. În Octave, realizați o funcție pentru a filtra un semnal de intrare x cu un filtru IIR în forma lattice, folosind coeficienții K și V :

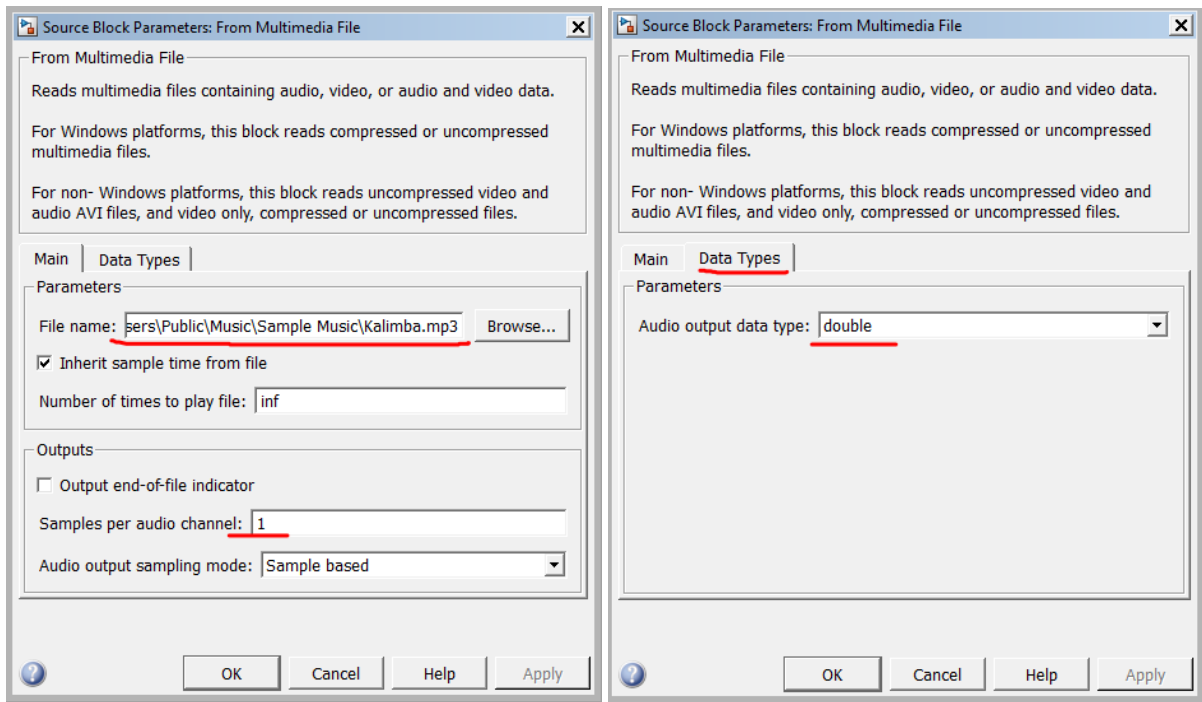
```
y = filter_latc_iir(K, V, x)
```

În funcție, definiți variabilele $w1$, $w2$, ... pentru a stoca valorile din celulele de întârziere, și $w1_next$, ... pentru valorile lor următoare.

- Calculați ieșirea curentă pe baza valorilor $w1$, ... și a intrării curente
 - Calculați valorile următoare $w1_next$, ... pe baza valorilor $w1$, ... și a intrării curente
 - Actualizați $w1$, ... cu valorile din $w1_next$, ... apoi se iterează din nou
6. Utilizați funcția de mai sus pentru a filtra un semnal audio.
- a) Încărcați fișierul folosind `audioread()`;
- b) Filtrați semnalul cu funcția `filter_latc_iir()` de mai sus, și afișați/redați semnalul obținut.

Observații:

- Veți avea nevoie de blocurile *Unit Delay*, *Sum* și *Gain*
- La intrare puneți un bloc *From Multimedia File*, la ieșire un bloc *To Audio Device*
- La ieșire, înainte de blocul *To Audio Device* intercalați un bloc *Manual Switch* la care semnalul original și semnalul filtrat, pentru a putea comuta ușor între cele două
- La blocul *From Multimedia File* selectați un fișier audio (de ex. Kalimba.mp3 din My Documents) și puneți setările *Sample-based*, *Samples per audio channel* = 1 și "DataTypes/Audio output data type" = *double*



- Setăți parametrii modelului Simulink pentru o simulare discretă, cu pas fix (auto):
 - Type: *Fixed-step*
 - Solver: *discrete (no continuous states)*

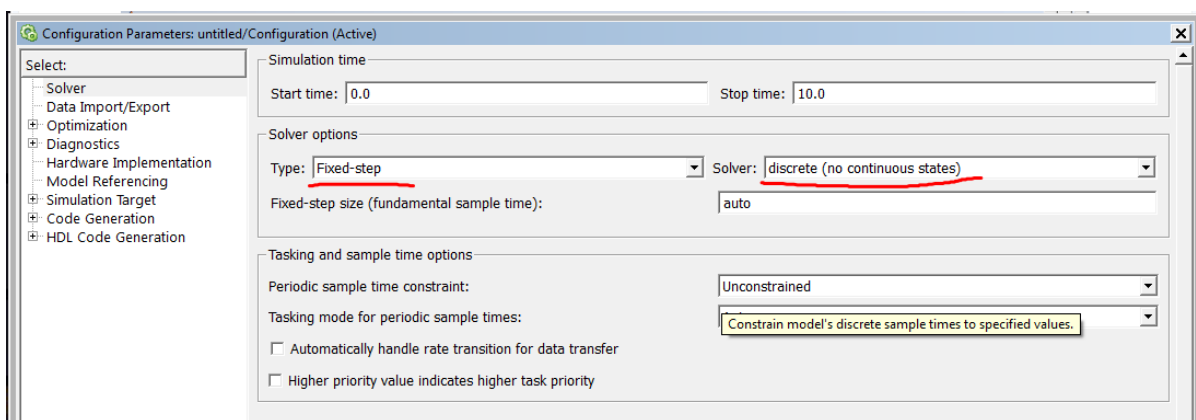


Figure 1: Model settings for discrete models

Întrebări finale

1. TBD